

AMENDMENTS TO THE SPECIFICATION

Please replace entire specification with the following amended specification:

Concurrent Web Based Multi-Task Support for Control Management System

By

Sheng (Ted) Tai Tsao

7/29/2002

Field of the Invention

The present invention generally relates to computer communications network. More specifically, the present invention relates to a web based support user to perform multiple simultaneous concurrent tasks or system operations from a single, or multiple web-based consoles (web console) over the central controlled distributed scalable virtual machine system ("CCDSVM"). This invention also relates to the creation of a web-based computer user work environment for a single or multiple computer systems. With this web-based computer user work environment the entire CCDSVM system can be operate like a single virtual machine.

Background of the Invention

Terminology:

CCDSVM:

It is an abbreviation for central controlled distributed scalable virtual machine. The CCDSVM allows a control management station to control a group of systems and provide distributed services to client system in Intranet and Internet as well as in LAN environment. The software components of CCDSVM form the virtual operating system.

Computer User Work Environment:

~~Every~~ A typical computer system provides ~~[[end users]]~~ a computer user work environment to end users, wherein the computer user work environment ~~[[, which usually]]~~ runs on top of a generic computer operating system. With this work environment, an end user ~~[[s are capable to]]~~ can login to the system and. ~~After login, users can setup various~~ computer resource~~[[s]]~~ access controls for each individual ~~[[users]]~~ preference based on his or her ~~their~~ permitted role. The users, for example, ~~also can configure computer resources~~ such as disks, networks, file folder/directory systems, and others, ~~and can run~~ Various computer tasks & operations can be executed by the ~~through~~ computer application, and let the computer operating system of the system provides to carry out the results of tasks to the end user. Specially, with a generic computer operating system, this work environment allows multiple concurrent users and each of them to run multiple concurrent tasks or operations simultaneously. The computer user work environment has evolved from paper tape & punch card environment, command line environment on a native system to window & mouse click environment on a native system in the past. This invention provides users a web-based computer user work (operating) environment on top of generic operating system for a single or multiple computers and to allows user work on one or multiple computer systems through a conventional web-browser.

Web-Console

~~When a server provides software support, which allows user access and performs~~ system operations on this server through a web browser on a system (desktop, laptop, server, PDA, or cell phone) anywhere on net, this special web browser of that system becomes a web-console. With CCDSVM, the permitted user from web-console should be able to access and operate entire CCDSVM.

~~Thread and Process:~~

~~To simplify the discussion, the term of thread and process are roughly used without differentiation between them in this invention regardless the very restricted definition of the thread and process in computer science field. Here both thread and process are basically referred as a sequence of instructions based on a piece of program code, which starts to be executed by a computer system step by step and further to carry out a computer task.~~

~~The lock:~~

~~Lock is a mechanism, with which a thread can use it to protect a computer resource from other threads, which attempt to manipulate this same resource at the same time. There is conventional lock, which can be acquired and released by same thread. The conventional lock mechanisms have used by most software developer crossing the software industry. The lock described in this invention may or may not be a conventional one. The non-conventional lock mechanisms created in this invention can be acquired by one thread and may be released by same thread or by another thread. Therefore, it is non-conventional lock serving threads on the native system.~~

Figures:

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated into and constitute a part of this specification, illustrate one or more examples of embodiments and, together with the description of example embodiments, serve to explain the principles and implementations of the embodiments.

In the drawings:

Fig. 1[[: An]] illustrates an example of a simplified multi-tasks support on Web-console in a simple environment.

Fig. 2[[: An]] illustrates an example of a simplified multi-tasks support on Web-console in a CCDSVM environment.

Fig. 3[[: An]] illustrates an example of basic data flow between a web-console on a console host and the console supporting software [[on]] in a control management environment.

Fig. 4[[: An]] illustrates an example of a task and an operation processing flow chart in a CCDSVM environment.

Fig. 5[[: The]] illustrates a user space task list, which [[is]] includes an abstraction of data structure for multiple simultaneous concurrent tasks and operations control in CCDSVM environment.

Fig. 6[[: The]] illustrates a layered CCDSVM structure.

~~In the drawing, like elements are designated by like reference numbers.~~

Brief Description of This Invention

The traditional web server or other server (3 of Fig. 1) may support users from a web browser (8 of Fig. 1) somewhere on the net to perform some tasks, which could get quick responses and could be finished in a short period of time such as checking web server's status or getting a server's other information, et cetera [[etc.]]. However, it may not support well for multiple simultaneous concurrent tasks or operations from the same web-browser, especially when these tasks are time consuming to ~~finish~~be finished. For example, ~~create~~ creating a 60GB file system on the server, or ~~configure~~ configuring a raid controller on that server (3 of Fig. 1) is a large (and time consuming) task. Because these tasks often take a large amount of time and will hang in the web-console on console host (1 of Fig. 1), as a result, no other tasks could be performed parallel from the same web-console at the same time. In addition, traditional console supporting software (5 of Fig. 1) does not work for a more complicated environment such as in a CCDSVM environment (Fig. 2).

To solve these problems and effectively to support multiple simultaneous concurrent tasks on a web-console for both simple and CCDSVM environment, the console supporting software (5 of Fig. 1) needs to ~~be expanded to~~ include additional control management software modules (4 of Fig. 2) and others. The control management software module shall communicate and control all system units (3 of Fig. 2). Also, each system units (3 of Fig. 2) needs service[[s]] software modules (8 of Fig. 2) to communicate with control management software (4 of Fig. 2) of console support software (6 of Fig. 2). In addition, a[[n]] user space task list (Fig. 5) could be used together with conventional or non-conventional locks to support all multiple simultaneous concurrent tasks and operations. With this invention, the multi-tasks support on the web-console in a simple environment (Fig 1) has been viewed as a special case of such support in a CCDSVM environment (Fig. 2). The CCDSVM (Fig. 2) will be degenerated into a simple server environment (Fig. 1) if multiple system units (3 of Fig. 2) do not present.

These and other features, aspects and advantages of the present invention will become understood with reference to the following description, appended claims and accompanying figures [[where:]] set forth below.

Brief Description of the Drawings

Detailed Description of the Invention

CCDSVM is configured to provide a control management station to control a group of systems and provide distributed services to at least one client system over an Intranet, an Internet as well as a LAN environment. The software components of CCDSVM form the virtual operating environment.

When a server provides software support, which allows a user access the server and performs system operations on this server, a web-browser on a system (such as a desktop, laptop, server, PDA, or cell phone), which can be anywhere on net, this special web-browser of that system becomes a web-console. With CCDSVM, a permitted user from web-console should be able to access and operate entire CCDSVM.

To simplify the discussion, the term of thread and process are roughly used without differentiation between them in this invention regardless the very restricted definition of the thread and process in computer science field. Here both thread and process are basically referred as a sequence of instructions based on a piece of program code, which starts to be executed by a computer system step by step and further to carry out a computer task.

Lock is a mechanism that allows a thread to lock a computer resource for its own use and prevents other threads, which attempt to manipulate this same resource at the same time, from access to the same computer resource. There is conventional lock, which can be acquired and released by the same thread. The conventional lock mechanisms have used by most software developer crossing the software industry. The lock described in this invention may or may not be a conventional one. The non-conventional lock mechanisms created in this invention can be acquired by one thread and may be released by same thread or by another thread. Therefore, it is non-conventional lock serving threads on the native system.

FIG. 1[[: Shows]] illustrates an example of a web-console scheme in a simple environment, which includes a console host 1, a server 3, and a net 2. Console host 1 further

includes a web-console 8 having a browser. Server 3 includes a web-console, a web server software 4, and a console support software 5, wherein said console support software 5 further includes web server interface 6 and other service modules 7. It should be noted that the same reference indicators will be used throughout the drawings and the following description to refer to the same or like items.

[[a]] Console host (1), from which a user is able to perform system tasks or operations for server (3) through web-console (browser) (8) of console host (1). ~~Console hosts (1) could~~ can be any system on the net 2, such as a server, a desktop PC, a laptop PC, a hand held PDA, or a cell phone. The web browser (8) may be ~~an exist~~ commercially available software from any vendor or a proprietary software companies. The web browser 8, ~~which~~ is able to handle the web protocol such as HTTP. The console host 1 may also include other software modules (9), which may be implemented with any suitable programming languages such as C, C++, Java, XML, et cetera. Other software modules 9 are used to ~~[[etc and]]~~ communicate ~~[[with]]~~ between said server (3) and said console host 1 using IP, non-IP or any suitable protocols ~~[[to get or send]]~~ for receiving and/or sending data between said console host 1 and said server (3).

[[b]] Server (3), which could be a web server or any kind of system with web server software. ~~The server consists~~ includes web server software (4) and console supporting software (5). The console supporting software (5) includes web server interface (6) and other services software modules (7), ~~[[which]]~~ wherein said other services software 7 operates on native server system. The web server software (4) may be ~~[[commercial]]~~ commercially available or proprietary software, which is able to accept and handle the web protocol such as HTTP. ~~It may also has a~~ A native web-console (browser) (8), therefore, enables a user also ~~can~~ to access and operate the computer locally.

e) ~~The net (2), which represent the~~ Net 2 represents a network infrastructure such as Internet and intranet, LAN with all kinds of related network equipment and media such as switch/router, different kind of cables and wireless media.

FIG. 2~~[[:]]~~ shows an example simplified block diagram of an embodiment of CCDSVM. The system includes console hosts 1, a control management station 2, system

units 3, a net1 11, a net2 12, and client systems 10. [[d]] Console hosts (1), ~~which~~ could be any system on the net such as a server, a desktop PC, a laptop PC, a hand held PDA, or a cell phone[[, where a]]. A web-console (web browser) (9) can be used to access and operate the entire CCDSVM. The web browser (9) may be a-existing commercially available software from any vendor or proprietary software company. The web browser 9, which is able to handle web protocol such as HTTP. The term of the A difference between a web-console 9 and a web browser is that the web-console 9 is capable of accessing used instead of using the term of web browser due to its ability to access system information and performing system operation in the CCDSVM environment. The console host may also include other software modules (13), which may be implemented with any suitable programming languages such as C, C++, Java, XML etc. These software modules may be used to facilitate communications between communicate with control management station (2) and console hosts 1 using IP, non-IP or any suitable protocols for receiving or sending to get or send data between console host and control management station (2). To support a non-web-based networked console, the software used [[for]] by the web console [[must]] 9 is configured to handle protocol other than web protocol (HTTP) and this software must communicate with console supporting software on control management station.

[[e]] Control management station (2): ~~It~~ could be any system on the net such as a server, a desktop PC, a laptop PC, or others. The control management station includes web server software (7) and console supporting software (6). The console supporting software (6) includes web server interface software modules (5) and control management software modules (4). It should be noted that the console supporting software 6, and it may include others service software modules. It may also have native Web browser used as a web-console (9) of native system. The web server software (7) discussed earlier in this invention could be an-existing commercially available software from a major vendor or other proprietary software company, which is able to accept and handle the web protocol such as HTTP. The web server software (7) sends data to and receives data from web-console 9 (9 of Fig. 2) of console hosts 1 (1 of Fig. 2).

The console supporting software (6) can be implemented with any suitable languages such as C, C++, Java, XML, etc. or even implemented by using a combination of different

languages as long as it provides the features and functionality described in this invention. That means it is language independent. In addition, the communication protocol used between console support software (6) and service software modules (8) of system units (3) could be any suitable protocol such IP based, or non-IP based or other protocols.

There may be several fixed threads being created based on control management software modules (4). There are may be various number of threads that are created based on web server interface software modules (5) for each tasks being initiated by a user at the web-console 9 (~~9 of Fig. 2~~). All of these threads may be communicated with each other through inter-process communication ~~[[both]]~~ based on web interface modules (5) and ~~[[based on]]~~ control management module (4), which are simply referred ~~[[as]]~~ to the thread of console supporting software (6). However, ~~sometime~~ to simplify the discussion, they may be just referred ~~to~~ ~~[[as]]~~ console supporting software (6) without mentioning the thread at all.

If there is a need ~~[[needs]]~~ to support a less effective non-web-based networked console, there is no need for web server software (7) and web server interfacing software module (5) to be employed. Instead, an additional network software module is required and it could be implemented with any suitable programming language and any suitable communication protocol other than web protocol (HTTP). This network software module can communicate with networked console software on console host (1) via network link and further to communicate with the rest of console supporting software (6) via inter-process communication mechanism.

~~[[f]]~~ System unit (3) ~~[[,which]]~~ could be any system on the net such as a server, a desktop PC, a laptop PC, a hand held PDA, ~~[[or]]~~ a cell phone, any operational system, a ~~device or component such as~~ video server, a web server, a storage block data server (SAN unit), a video monitoring device, and so forth others. The system unit contains service software modules (8), which ~~[[is]]~~ are capable of communicating ~~to communicate~~ with the outside ~~[[of]]~~ world. For example, service modules 8 is used to communicate with control management software (4) of control management station (2) ~~[[to carry]]~~ for carrying out the tasks, or to communicate with clients 10 of CCDSVM (~~10 of Fig. 2~~) ~~[[to deliver]]~~ for delivering the services to them, or to communicate with another system unit (3) ~~[[to transfer]]~~

for transferring the data. The service software modules could be implemented with any suitable programming languages such as C, C++, Java, or others~~[[, and]]~~. It should be noted that the communication protocol could be any suitable protocol such as IP (Internet Protocol) base or other non-IP based protocol.

~~[[g]]~~ The net1 (11), ~~which~~ represents any kind of communication link between control management station (2) and web-console (9) or client hosts (10). The link could be an infrastructure of internet, intranet, LAN, or others. The link ~~[[and]]~~ could use connection media such as cable (Ethernet, optical Fibre, and other), wireless media, bus. The link ~~[[, and]]~~ could also use communication equipment such as switches, routers, and/or adapters ~~switch/routers/adapters, etc.~~

~~[[h]]~~ The net2 (12), ~~which~~ represents any kind of communication link between control management station (2) and the system units (3) or web-consoles (9). The link could be an infrastructure of internet, intranet, LAN, or other. The link ~~[[and]]~~ could use connection media such as cable (Ethernet, optical Fibre, and other), wireless media, bus, and could use communication equipment such switch/routers/adapters etc.

~~[[i]]~~ Client systems (10). ~~The client systems~~ are not part of CCDSVM~~[[]]~~ but they may requests services from CCDSVM, as shown in ~~[[]]~~Fig. 2~~[[]]~~. The role of client systems and the connections between client system (10) and system unit (3) will not be described in this invention since they are irrelevant to ~~[[in]]~~ this invention.

FIG. 3: ~~Shows the~~ illustrates a simplified data flow between web-console (9 of Fig. 2) on console host (1 of Fig. 2) and the console supporting software (6 of Fig. 2) on control management station (2 of Fig. 2). Data travels from web-console (9 of Fig. 2) to console supporting software (6 of Fig. 2) includes two steps. First, data goes from web-console (9 of Fig. 2) to web server software (7 of Fig. 2) via net (11 or 12 of Fig. 2). Second, console supporting software (6 of Fig. 2) gets data from web server software (7 of Fig. 2) via inter-process communication. To simplify the discussion of this invention, this data traveling path will simply refer to as console supporting software (6 of Fig. 2) getting data from web-console (9 of Fig. 2) or refer to as the data being ~~[[send]]~~ sent from web-console (9 of Fig. 2) to the console supporting software (6 of Fig. 2).

Data travels from console supporting software (6 of Fig. 2) to web-console (9 of Fig. 2) includes two reverse steps. First, the web server software (7 of Fig. 2) gets data from console supporting software (6 of Fig. 2) via inter-process communication. Second, the web server software (7 of Fig. 2) sends data to web-console (9 of Fig. 2) via net (11 or 12 of Fig. 2). To simplify the rest discussion in this invention, this reverse data traveling will refer to as data being sent from console supporting software (6 of Fig. 2) to web-console (9 of Fig. 2) or refer to as web-console (9 of Fig. 2) gets data from console supporting software (6 of Fig. 2). ~~Here the data may also be referenced as information or information on a web page etc.~~ The terms data, information, and information on a web page will be used interchangeably herein.

FIG. 4[[:]] shows the basic tasks and operation processing flow chart, which initiated from web-console.

FIG. 5[[:]] shows [[the]] a user space task list. Each slot on the user space task list can be used to hold [[a]] task information relating to a task issued from Web-console.

FIG. 6[[: Shows the]] illustrates a layered CCDSVM environment, which provides a flexible scalability mechanism to efficiently support thousands of heterogeneous system units. With this structure, a control management station (2 of Fig. 2), which is situated at middle layer 2 [[also]] becomes a system unit[[s]] of the up layer1.

FIG. 7[[: Shows the]] illustrates typical hardware components [[of]] including a control management system, system units, and console hosts. It consists of one or ~~multiple~~ more CPU, memory, secondary storage such as disks or memory sticks, the network interface cards, and display components such as monitor or others. These components are connected internally through buses.

Detailed Description of the Invention

The detailed explanation of Fig. 2 will show that how multiple concurrent tasks can be initiated from a web-console (9 of Fig. 2) and [[to]] can be executed either on any one of the system units (3 of Fig. 2) or on a control management station (2 of Fig. 2) according to this invention.

In one example, a user A ~~[[from]]~~ at a web-console (9 of Fig. 2) receives an authentication from the ~~got authenticated by~~ console supporting software (6 of Fig. 2). In one embodiment, a ~~such as~~ successfully login on a control management station (3 of Fig. 2) is considered an authentication because it authorizes user A to access the system. So that user A has obtained necessary information ~~[[of]]~~ about all system units (3 of Fig. 2) and control management station (2 of Fig. 2) from console supporting software (6 of Fig. 2). When user A ~~[[initiated]]~~ initiates a task for a selected target system, which is either a system unit (3 of Fig. 2) or the control management station (2 of Fig. 2), the task information is transmitted via net (11 or 12 of Fig. 2) from web-console (9 of Fig. 2) to the console support software (6 of Fig. 2) on control management station (2 of Fig. 2). A thread is created based on console support software (6 of Fig. 2) wherein the thread ~~and it~~ will serve and carry out this task. The created thread stores the task information into a valid entry ~~[[in]]~~ on a user level task list (Fig. 5) and obtains related locks. This ensures that the multiple tasks can be initiated simultaneously within ~~[[a]]~~ the same web-console (9 of Fig. 2) without delaying, effecting or blocking ~~[[each]]~~ other potential thread(s). In addition, multiple web-consoles for multiple simultaneous users anywhere on the net (11 or 12 of Fig. 2) also can be supported. The obtained locks for this task will be properly released one at a time along with the task execution up to ~~[[the]]~~ a point when the task is finally ~~[[got]]~~ finished. Therefore, each task~~[[s]]~~ could be executed without time delay.

If the total tasks initiated from web-console (9 of Fig. 2) have succeeded the maximum allowed tasks by console supporting software (6 of Fig. 2), the initiated task is failed. The locks will be released by the corresponding thread and the user A on web-console (9 of Fig. 2) is notified correspondingly via net (11 or 12 of Fig. 2).

If an existing task is in ~~[[the]]~~ a stage of changing a resource object on a target system and if ~~[[the]]~~ a newly created task will change ~~operate~~ on the same resource object on that target system, the newly initiated task may will fail or may will have to wait until the previous task ~~to-be~~ is finished. Further, if a task is failed, the locks associated with the task will be release by the thread and the user A on the web-console (9 of Fig. 2) will be notified via net (11 or 12 of Fig. 2) by console support software (6 of Fig. 2).

The credential of executing a specific task on a specific target system by user A is checked. In addition, the ordinary users' access & operation permissions and credentials are setup by the administrator of the system with supervisor user or special privileges. If user A is not permitted to perform any task on such target system or is not permitted to perform such task on any system, the task execution will ~~[[be]]~~ fail and user A will be notified via net (11 or 12 of Fig. 2). Otherwise, the task will be carried out by the corresponding thread on the target system, which is either a control management station (2 of Fig. 2) or a system unit (3 of Fig. 2). If there is a need~~[[s]]~~, the console supporting software (6 of Fig. 2) will send the results (or data) back to web-console (9 of Fig. 2). When a task is neither failed nor ~~either failed or succeeded~~, the threads of console supporting software (6 of Fig. 2) will release the locks acquired for this task.

If the task needs to be executed on the control management station (2 of Fig. 2), the thread created based on console supporting software (6 of Fig. 2) will carry out this task. The threads of console support software (6 of Fig. 2) also need to determine if ~~it needs~~ they need to create another thread to execute this task. If there is a need~~[[s]]~~, another thread will be created to execute this task. Once the task is finished, the corresponding locks will be released by console supporting software (6 of Fig. 2).

If a task needs to be executed on a system unit (3 of Fig. 2), the console supporting software (6 of Fig. 2) will transmit the task information via net2 (12 of Fig. 2) to the service software module (8 of Fig. 2) of target system unit (3 of Fig. 2). The thread based on service software module (8 of Fig. 2) of target system unit (3 of Fig. 2) will carry out this task. The service software module (8 of Fig. 2) on target system unit (3 of Fig. 2) needs to determine if an additional thread needs to be created in order to execute such task. If there is a need, an additional thread is created to execute this task. Once the task is finished on the target system unit (3 of Fig. 2), the corresponding status of the task execution is transmitted back to the console supporting software (6 of Fig. 2) of the control management station (2 of Fig. 2). Upon receiving the task finished status, the locks associated with the thread of console support software (6 of Fig. 2) for that task are released.

The Task Issued From Web-Console~~[[:]~~

The multiple concurrent tasks issued from a web-console (9 of Fig. 2) by a user could be any of the followings:

- a) Move or transmit data such as a multiple [[Gig Bytes]] gigabytes of file or other data in any form from any point or any system to another point or system within CCDSVM (Fig. 2).
- b) Configure, partition and assign entire storage system (raid/disk) within CCDSVM (Fig. 2).
- c) Setup authentication of specific user from a specific web-console (1 of Fig. 2) with certain privilege for entire CCDSVM or for a specific system, which could be any system unit (3 of Fig. 2) or a control management station (2 of Fig. 2). [[Setup]] Setting up the steps of authentication process for any specific services on one or more specific system units can also be implemented.
- d) Monitor and display activities and status for networks, storage, CPUs, processes and threads ~~activities and status for entire~~ in CCDSVM.
- e) ~~Creating~~ Create file system, file and directory structures, and all other related data file operations on either the control management system (2 of Fig. 2) or system units (3 of Fig. 2).
- f) And all other kind of tasks and operations, which might be run in other OS (operating system) environment.

The capability of providing user with the multiple concurrent simultaneous operations and tasks on web console (9 of Fig. 2) has indicated that this invention has created a web-based user work environment on top of an existing operating system of a single or multiple computer systems. Further, this is a consistent working environment for a computer system since it allows a user to access exact the same working environment through the web-console (9 of Fig. 2), which could be a web browser either on a native system or from other remote systems.

User Login[[:]]

The user-login mechanism is also supported by the console supporting software (6 of Fig. 2). The web-console (9 of Fig. 2) ~~[[obtained]]~~ obtains a login web page from the console supporting software (6 of Fig. 2) via the net (11 & 12 of Fig. 2). Once the user provides an account name and a password for the login page on the web-console (9 of Fig. 2), the authentication information is sent to the console supporting software (6 of Fig. 2) for validation. Upon ~~succcess-of~~ successful validating the user account and password information, the console support software (6 of Fig. 2) sends ~~[[the]]~~ all necessary system information such as IP address ~~[[etc]]~~ to web-console (9 of Fig. 2), which includes the information of control management station (2 of Fig. 2) and system units (3 of Fig. 2).

The Maximum Tasks~~[[:]]~~

The maximum multiple concurrent simultaneous tasks that can be initiated from web-consoles are determined by console support software modules (6 of Fig. 2), and they are also determined based on the needs and the capacity of the control management station (2 of Fig. 2).

The Credential Checking~~[[:]]~~

The credential of a user includes the permission to access all or partial systems, or a single system within CCDSVM. The credential further includes~~[[,]]~~ the permission to run all tasks or partial tasks, which lists the ~~which listed in~~ previous section of "The Task Issued From Web-Console". It also includes the permission of accessing specific size of storage volumes~~[[etc]]~~. For example, user B may be granted a permission to run tasks on systems X, Y, and Z. Another user C may be granted a permission to run tasks on entire system in the CCDSVM environment. User C might be allow to get system status on systems X, Y, and Z only while user B allows to run all tasks on systems X, Y, and Z. Each system mentioned here could be a control management station or any of system units. This basically represents a two-level~~[[s of the]]~~ authentication policy and checking. The first level is the security imposed ~~[[for]]~~ by the control management station (2 of Fig. 2) and the second level is the security imposed ~~[[for]]~~ by system units (3 of Fig. 2).

The Web-Server Interface Software Module~~[[:]]~~

The web-server interfacing software module (5 of Fig. 2) is responsible to get information from or send information to web server software (7 of Fig. 2). It also interacts with control management modules (4 of Fig. 2) via inter-process communication and communicates with service module (8 of Fig. 2) of system unit (3 of Fig. 2) via net2 (12 of Fig. 2).

The Control Management Software Module[[:]]

The control management module (4 of Fig. 2) on the control management station (2 of Fig. 2) is responsible for communicating with system units (3 of Fig. 2). It sends data to or receives data from system units via net2 (12 of Fig. 2). It also provides information of system units (3 of Fig. 2) to web interface software modules (5 of Fig. 2) of the control management station (2 of Fig. 2) via an inter-process communication mechanism.

The Layered CCDSVM Structure[[:]]

To be more efficiently support multiple concurrent tasks over a larger number of system units (3 of Fig. 2), the CCDSVM can be organized into ~~multiple layers~~ a multi-layer structure (Fig. 6). With this layered structure, the CCDSVM can be sub-divided into different groups. For example, each level-2 control management station could function as both control management station (2 of Fig. 2) to the system units below it and could function as a system unit (3 of Fig. 2) for level-1 control management station. Therefore, it must have related software modules for both the control management station and system units.